LISTING OF THE CLAIMS

The following listing, if entered, replaces all prior versions of the claims in the present application.

1. (Currently Amended) A method comprising:

receiving at least one packet; [[and]]

~~disposing of the received at least one packet in response to a walk of~~ accessing a Hash Table, based upon information included in the at least one packet, wherein

the Hash Table is balanced,

the Hash Table ~~is configured to~~ stores Binary Comparison Trees, and

the Hash Table ~~is configured to~~ encodes an Access Control List; and

disposing of the at least one packet, based upon the Access Control list encoded in the Hash Table.

2. (Currently Amended) The method of Claim 1, wherein said disposing of the received at least one packet ~~in response to a walk of the Hash Table~~ further includes:

constructing a hash table index value from one or more bit positions, within the received at least one packet, pointed at by one or more pointers of a Hash-Table-Balancing Bit Selection Vector; and

walking a binary comparison tree, stored within the Hash Table, associated with the constructed hash table index value.

3. (Previously Presented) The method of Claim 1, further comprising:

converting the Access Control List to the Hash Table.

4. (Previously Presented) The method of Claim 3, wherein said converting the Access Control List to the Hash Table further includes:

creating a binary comparison tree for at least one Access Control List rule in the Access Control List.

5. (Original) The method of Claim 4, wherein said creating a binary comparison tree for at least one Access Control List rule further includes:

creating at least one node, having at least one miss branch and at least one match branch, for at least one packet header field utilized by the at least one Access Control List Rule in the Access Control List.

6. (Previously Presented) The method of Claim 3, wherein said converting the Access Control List to the Hash Table further includes:

inserting at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index.

7. (Original) The method of Claim 6, wherein said inserting at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index further includes:

generating a hash table index value for the at least one Access Control List rule; and

inserting the at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value.

8. (Original) The method of Claim 7, wherein said inserting the at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value further includes:

inserting, in its entirety, the binary comparison tree constructed for the at least one Access Control List rule into the hash table entry pointed at by the hash table index in response to a determination that no pre-existing binary comparison tree is resident within the hash table entry.

9. (Original) The method of Claim 7, wherein said inserting the at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value further includes:

inserting at least one node of the binary comparison tree constructed for the at least one Access Control List rule into the hash table entry pointed at by the hash table index in response to a determination that a pre-existing binary comparison tree is resident within the hash table entry.

10. (Original) The method of Claim 7, wherein said generating a hash table index value for the at least one Access Control List rule further includes:

constructing the hash table index value from the contents of one or more packet headers utilized by the at least one Access Control List rule in the Access Control List.

11. (Original) The method of Claim 10, wherein said constructing the hash table index value from the contents of one or more packet headers utilized by the at least one Access Control List rule in the Access Control List further includes:

constructing the hash table index value from the contents of the one or more packet header bit positions pointed at by one or more pointers of a Hash-Table-Balancing Bit Selection Vector.

12. (Original) The method of Claim 11, wherein said constructing the hash table index value from the contents of the one or more packet header bit positions pointed at by one or more pointers of a Hash-Table-Balancing Bit Selection Vector further includes:

constructing the Hash-Table-Balancing Bit Selection Vector.

13. (Original) The method of Claim 12, wherein said constructing the Hash-Table-Balancing Bit Selection Vector further includes:

defining one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the Access Control List.

14. (Original) The method of Claim 13, wherein said defining one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the Access Control List further includes:

defining the one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions, which appear relatively most frequently, within the one or more packet header fields utilized by the one or more Rules of the Access Control List.

15. (Original) The method of Claim 13, wherein said defining one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the Access Control List further includes:

defining the one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions, whose contents have relatively equal variation between logical one and logical zero, within the one or more packet header fields utilized by the one or more Rules of the Access Control List.

16. (Currently Amended) A system comprising:

a Hash Table;

means for receiving at least one packet; and

means for disposing of the received at least one packet, based upon an Access Control list encoded in the ~~in response to a walk of a~~ Hash Table, wherein the means for disposing of the at least one packet are configured to access the Hash Table, based upon information included in the at least one packet, and wherein

the Hash Table is balanced,

the Hash Table ~~is configured to~~ stores Binary Comparison Trees, and

the Hash Table ~~is configured to~~ encodes the [[an]] Access Control List.

17. (Currently Amended) The system of Claim 16, wherein said means for disposing of the received at least one packet ~~in response to a walk of a Hash Table~~ further includes:

means for constructing a hash table index value from one or more bit positions, within the received at least one packet, pointed at by one or more pointers of a Hash-Table-Balancing Bit Selection Vector; and

means for walking a binary comparison tree associated with the constructed hash table index value.

18. (Previously Presented) The system of Claim 16, further comprising:

means for converting the Access Control List to the Hash Table.

19. (Previously Presented) The system of Claim 18, wherein said means for converting the Access Control List to the Hash Table further includes:

means for creating a binary comparison tree for at least one Access Control List rule in the Access Control List.

20. (Original) The system of Claim 19, wherein said means for creating a binary comparison tree for at least one Access Control List rule further includes:

means for creating at least one node, having at least one miss branch and at least one match branch, for at least one packet header field utilized by the at least one Access Control List rule in the Access Control List.

21. (Previously Presented) The system of Claim 18, wherein said means for converting the Access Control List to the Hash Table further includes:

means for inserting at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index.

22. (Original) The system of Claim 21, wherein said means for inserting at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index further includes:

means for generating a hash table index value for the at least one Access Control List rule; and

means for inserting the at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value.

23. (Original) The system of Claim 22, wherein said means for inserting the at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value further includes:

means for inserting, in its entirety, the binary comparison tree constructed for the at least one Access Control List Rule into the hash table entry pointed at by the hash table index in response to a determination that no pre-existing binary comparison tree is resident within the hash table entry.

24. (Original) The system of Claim 22, wherein said means for inserting the at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value further includes:

means for inserting at least one node of the binary comparison tree constructed for the at least one Access Control List rule into the hash table entry pointed at by the hash table index in response to a determination that a pre-existing binary comparison tree is resident within the hash table entry.

25. (Original) The system of Claim 22, wherein said means for generating a hash table index value for the at least one Access Control List rule further includes:

means for constructing the hash table index value from the contents of one or more packet headers utilized by the at least one Access Control List rule in the Access Control List.

26. (Original) The system of Claim 25, wherein said means for constructing the hash table index value from the contents of one or more packet headers utilized by the at least one Access Control List rule in the Access Control List further includes:

means for constructing the hash table index value from the contents of the one or more packet header bit positions pointed at by one or more pointers of a Hash-Table-Balancing Bit Selection Vector.

27. (Original) The system of Claim 26, wherein said means for constructing the hash table index value from the contents of the one or more packet header bit positions pointed at by one or more pointers of a Hash-Table-Balancing Bit Selection Vector further includes:

means for constructing the Hash-Table-Balancing Bit Selection Vector.

28. (Original) The system of Claim 27, wherein said means for constructing the Hash-Table-Balancing Bit Selection Vector further includes:

means for defining one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the Access Control List.

29. (Original) The system of Claim 28, wherein said means for defining one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the Access Control List further includes:

means for defining the one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions, which appear relatively most frequently, within the one or more packet header fields utilized by the one or more Rules of the Access Control List.

30. (Original) The system of Claim 29, wherein said means for defining one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the Access Control List further includes:

means for defining the one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions, whose contents have relatively equal variation between logical one and logical zero, within the one or more packet header fields utilized by the one or more Rules of the Access Control List.

31. (Previously Presented) The system of Claim 16, further comprising:

signal bearing media bearing

said means for receiving at least one packet, and

said means for disposing of the received at least one packet in response to a walk of the Hash Table.

32. (Original) The system of Claim 31, wherein said signal bearing media further includes:

recordable media.

33. (Original) The system of Claim 31, wherein said signal bearing media further includes:

transmission media.

34. (Original) The system of Claim 16, wherein the system further includes:

a network station.

35. (Currently Amended) A program product comprising:

~~signal bearing media~~ a computer readable storage medium comprising a Hash Table and program instructions executable to: ~~bearing~~

~~means for receiving~~ receive at least one packet[[,]] ;

access ~~means for disposing of the received at least one packet in response to a walk of a~~ the Hash Table, based upon information included in the at least one packet, wherein

the Hash Table is balanced,

the Hash Table ~~is configured to~~ stores Binary Comparison Trees, and

the Hash Table ~~is configured to~~ encodes an Access Control List; and

dispose of the at least one packet, based upon the Access Control list encoded in the Hash Table.

36. (Original) The program product of Claim 35, wherein said signal bearing media further includes:

recordable media.

37. (Canceled)

38. (Currently Amended) The program product of Claim 35, wherein the program instructions are executable to ~~said means for disposing of the received at least one packet in response to a walk of a Hash Table further includes~~:

~~means for~~ construct[[ing]] a hash table index value from one or more bit positions, within the received at least one packet, pointed at by one or more pointers of a Hash-Table-Balancing Bit Selection Vector; and

~~means for~~ walk[[ing]] a binary comparison tree associated with the constructed hash table index value.

39. (Currently Amended) The program product of Claim 35, wherein the program instructions are executable to ~~said signal bearing media bears~~:

~~means for~~ convert[[ing]] the Access Control List to the Hash Table.

40. (Currently Amended) The program product of Claim 39, wherein the program instructions are executable to ~~said means for converting the Access Control List to the Hash Table further includes~~:

~~means for~~ creat[[ing]]e a binary comparison tree for at least one Access Control List rule in the Access Control List.

41. (Currently Amended) The program product of Claim 40, wherein the program instructions are executable to ~~said means for creating a binary comparison tree for at least one Access Control List rule further includes~~:

~~means for~~ creat[[ing]]e at least one node, having at least one miss branch and at least one match branch, for at least one packet header field utilized by the at least one Access Control List rule in the Access Control List.

42. (Currently Amended) The program product of Claim 39, wherein the program instructions are executable to ~~said means for converting the Access Control List to the Hash Table further includes~~:

~~means for~~ insert[[ing]] at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index.

43. (Currently Amended) The program product of Claim 42, wherein the program instructions are executable to ~~said means for inserting at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index further includes~~:

~~means for~~ generat[[ing]]e a hash table index value for the at least one Access Control List rule; and

~~means for~~ insert[[ing]] the at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value.

44. (Currently Amended) The program product of Claim 43, wherein the program instructions are executable to ~~said means for inserting the at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value further includes~~:

~~means for~~ insert[[ing]], in its entirety, the binary comparison tree constructed for the at least one Access Control List Rule into the hash table entry pointed at by the hash table index in response to a determination that no pre-existing binary comparison tree is resident within the hash table entry.

45. (Currently Amended) The program product of Claim 43, wherein the program instructions are executable to ~~said means for inserting the at least a part of a binary comparison tree constructed for at least one Access Control List rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value further includes~~:

~~means for~~ insert[[ing]] at least one node of the binary comparison tree constructed for the at least one Access Control List rule into the hash table entry pointed at by the hash table index in response to a determination that a pre-existing binary comparison tree is resident within the hash table entry.

46. (Currently Amended) The program product of Claim 43, wherein the program instructions are executable to ~~said means for generating a hash table index value for the at least one Access Control List rule further includes~~:

~~means for~~ construct[[ing]] the hash table index value from the contents of one or more packet headers utilized by the at least one Access Control List rule in the Access Control List.

47. (Currently Amended) The program product of Claim 46, wherein the program instructions are executable to ~~said means for constructing the hash table index value from the contents of one or more packet headers utilized by the at least one Access Control List rule in the Access Control List further includes~~:

~~means for~~ construct[[ing]] the hash table index value from the contents of the one or more packet header bit positions pointed at by one or more pointers of a Hash-Table-Balancing Bit Selection Vector.

48. (Currently Amended) The program product of Claim 47, wherein the program instructions are executable to ~~said means for constructing the hash table index value from the contents of the one or more packet header bit positions pointed at by one or more pointers of a Hash-Table-Balancing Bit Selection Vector further includes~~:

~~means for~~ construct[[ing]] the Hash-Table-Balancing Bit Selection Vector.

49. (Currently Amended) The program product of Claim 48, wherein the program instructions are executable to ~~said means for constructing the Hash-Table-Balancing Bit Selection Vector further includes~~:

~~means for defining~~ define one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the Access Control List.

50. (Currently Amended) The program product of Claim 49, wherein the program instructions are executable to ~~said means for defining one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the Access Control List further includes~~:

~~means for defining~~ define the one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions, which appear relatively most frequently, within the one or more packet header fields utilized by the one or more Rules of the Access Control List.

51. (Currently Amended) The program product of Claim 50, wherein the program instructions are executable to ~~said means for defining one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the Access Control List further includes~~:

~~means for defining~~ define the one or more pointers of the Hash-Table-Balancing Bit Selection Vector to point to one or more bit positions, whose contents have relatively equal variation between logical one and logical zero, within the one or more packet header fields utilized by the one or more Rules of the Access Control List.

52. (Currently Amended) A network station comprising:

~~a per-packet processing engine; and~~

a hash table, wherein

the hash table is balanced,

the hash table ~~is configured to~~ stores binary comparison trees, and

the hash table ~~is configured to~~ encodes an access control list[[,]];

an interface configured to receive a packet; and [[the]] a per-packet processing engine coupled to the hash table and [[is]] configured to [[walk]] access the hash table, based upon information included in the packet received by the interface,~~ in response to the network station receiving a packet, and~~, wherein the per-packet processing engine is configured to dispose of the at least one packet, based upon the Access Control list encoded in the Hash Table ~~the walk of the hash table identifies a disposition of the packet, according to the access control list encoded in the hash table~~.

53. (Previously Presented) The network station of Claim 52, wherein the per-packet processing engine is configured to:

construct a hash table index value from one or more bit positions, within the packet, pointed at by one or more pointers of a hash-table-balancing bit selection vector; and

walk a binary comparison tree, stored in the hash table, associated with the constructed hash table index value.

54. (Currently Amended) The network station of Claim 52, further comprising:

program instructions executable by the network station to convert the access control list to the hash table.

55. (Currently Amended) The network station of Claim 54, wherein the program instructions are is further executable by the network station to:

create a binary comparison tree for at least one access control List rule in the Access Control List.

56. (Currently Amended) The network station of Claim 54, wherein the program instructions are further executable by the network station to:

insert at least a part of a binary comparison tree constructed for at least one access control list rule into a hash table entry pointed at by a hash table index.

57. (Currently Amended) The network station of Claim 56, wherein the program instructions are further executable by the network station to ~~inserting at least a part of a binary comparison tree constructed for at least one access control list rule into a hash table entry pointed at by a hash table index includes~~:

~~generating~~ generate a hash table index value for the at least one access control list rule; and

insert[[ing]] the at least a part of a binary comparison tree constructed for at least one access control list rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value.

58. (Currently Amended) The network station of Claim 57, wherein the program instructions are further executable by the network station to ~~inserting the at least a part of a binary comparison tree constructed for at least one access control list rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value includes~~:

insert[[ing]], in its entirety, the binary comparison tree constructed for the at least one access control list rule into the hash table entry pointed at by the hash table index in response to a determination that no pre-existing binary comparison tree is resident within the hash table entry.

59. (Currently Amended) The network station of Claim 57, wherein the program instructions are further executable by the network station to ~~inserting the at least a part of a binary comparison tree constructed for at least one access control list rule into a hash table entry pointed at by a hash table index which is equal to the generated hash table index value includes~~:

insert[[ing]] at least one node of the binary comparison tree constructed for the at least one access control list rule into the hash table entry pointed at by the hash table index in response to a determination that a pre-existing binary comparison tree is resident within the hash table entry.

60. (Currently Amended) The network station of Claim 57, wherein the program instructions are further executable by the network station to ~~generating a hash table index value for the at least one access control list rule further includes~~:

construct[[ing]] the hash table index value from the contents of one or more packet headers utilized by the at least one access control list rule in the access control list.

61. (Currently Amended) The network station of Claim 60, wherein the program instructions are further executable by the network station to ~~constructing the hash table index value from the contents of one or more packet headers utilized by the at least one access control list rule in the access control list includes~~:

construct[[ing]] the hash table index value from the contents of the one or more packet header bit positions pointed at by one or more pointers of a hash-table-balancing bit selection vector.

62. (Currently Amended) The network station of Claim 62, wherein the program instructions are further executable by the network station to ~~said constructing the hash table index value from the contents of the one or more packet header bit positions pointed at by one or more pointers of a hash-table-balancing bit selection vector includes~~:

~~defining~~ define one or more pointers of the hash-table-balancing bit selection vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the access control list.

63. (Currently Amended) The network station of Claim 62, wherein the program instructions are further executable by the network station to ~~defining one or more pointers of the hash-table-balancing bit selection vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the access control list includes~~:

~~defining~~ define the one or more pointers of the hash-table-balancing bit selection vector to point to one or more bit positions, which appear relatively most frequently, within the one or more packet header fields utilized by the one or more rules of the access control list.

64. (Currently Amended) The network station of Claim 62, wherein the program instructions are further executable by the network station to ~~defining one or more pointers of the hash-table-balancing bit selection vector to point to one or more bit positions in one or more packet header fields utilized by one or more rules of the access control list includes~~:

~~defining~~ define the one or more pointers of the hash-table-balancing bit selection vector to point to one or more bit positions, whose contents have relatively equal variation between logical one and logical zero, within the one or more packet header fields utilized by the one or more rules of the access control list.